Paper ID #27348

How to Cultivate Computational Thinking-Enabled Engineers: A Case Study on the Robotics Class of Zhejiang University

Dr. Jingshan Wu, Zhejiang University

Postdoctoral Fellow of Institute of China's Science, Technology and Education Strategy, Zhejiang University; Lecturer, School of Public Administration, Zhejiang University of Finance & Economics

Ms. Yujie Wang, Zhejiang University

Postgraduate of Institute of China's Science,Technology and Education Strate, Zhejiang University

Ms. Hanbing Kong, Zhejiang University

Hanbing Kong, PhD Deputy Director, the Research Center for S&T, Education Policy, and Associate Professor of Management, Zhejiang University, and JEE Liaison for Research in Higher Education of Engineering.

Prof. Ling Zhu, Zhejiang University

Professor of Institute of China's Science,Technology and Education Strategy;Deputy Director of the Institute of Engineering Education, Zhejiang University.

How to Cultivate Computational Thinking-Enabled Engineers: A Case Study on the Robotics Class of Zhejiang University

Abstract

Computational Thinking (CT) is typically construed as an essential competence in solving problems and designing complex systems in the digital world. Robotics programs provide learning environments for acquiring core computational thinking skills. This study first proposes a framework of computational thinking in the context of engineering (CT-ENG), using qualitative content analysis on industry interviews. The authors then introduce the program of the Robotics Class of Zhejiang University in China, providing an integrative approach to teaching computational thinking effectively. The Robotics Class engages students in project-based computing-aided engineering activities throughout the four-year bachelor's program, and improves their computational thinking skills through engineering engagement. The findings in this study could have some implications for non-CS engineering majors to promote computing education and equip students with computational thinking at digital era.

Key Words: Computational Thinking, Engineering with Big E, Robotics, Case Study

I. Introduction

“To out-compete is to out-compute”, the U.S. Council on Competitiveness (2009) claimed for the computational endeavors 10 years ago. Computational thinking (CT) is defined as a way of solving modern engineering problems and designing complex systems that draws on concepts fundamental to computer science (Wing, 2006). Though CT has been discussed and debated for quite a long time, CT literature is still at an early stage of maturity (Filiz, 2016), especially how to teach computational thinking in the context of engineering is still challenging (Hacker M, 2018).

Integrating computational thinking (CT) into engineering programs offers a rich environment for scholarly research. Possible research questions might include:

RQ1. What are core computational thinking skills in the context of engineering?

RQ2. How to integrate computing in engineering curriculum so as to help engineering students learn computational thinking skills?

RQ3. How do engineering students learn computational thinking skills through the effective engagement in instructional activities?

RQ4. What should we endeavor to promote computational thinking for non-CS engineering majors?

II. Computational Thinking in Engineering

1. Computational Thinking (CT)

Computing is an innate capacity of human beings. The term Computational Thinking (CT) has been used in the educational context for quite a long time (Dijkstra, 1976). But the concept has become popular in recent years since Jeannette M. Wing (2006) represented it as “universally applicable attitude and skill set everyone, not just computer scientists”. After that, researchers redefined CT in different aspects, such as the process of problem solving (CSTA & ISTE, 2011; Kalelioglu, 2016) , the thinking attributes of computing (Chen G.L., 2012; Wenchong et al., 2014; Korkmaz Ö, et al., 2017) and common body of knowledge for computing (Zhan D.C. & Nie L.S., 2013).

Despite the unified definition, much of the literature focuses on CT as it applies to the disciplines, especially in STEM fields (Malyn-Smith & Lee, 2012; Weintrop, et al.,2014; Swaid,2015; Beheshti, et al., 2017). There is evidence that computational thinking is indispensible in modern engineering (Mohtadi,2013; Gross,2014), and CT could obviously develop and enhance students' academic competitiveness, problem-solving skills and awareness of complex systems (Magana, et al., 2013; Repenning, et al., 2010). Thus, computational thinking in the context of engineering and technological challenges, involves solving complex problems and designing systems by exploring and applying computational approaches.

2. Engineering with a Big E

"Engineering with a Big E" represents an education philosophy in the context of engineering for future, which is derived from MIT. The former President of MIT alerted that "engineering education must be more closely back to the fundamentals of engineering practice" (Vest, C., 1994), revealing the science-oriented educational paradigm for engineering can no longer be adapt for the emergence of complex systems and grand challenges. As a response, School of Engineering launched a new Long Range Plan "Engineering with a Big E: Integrated Engineering Education" (Moses, J., 1994), which is essentially the integration of engineering technologies and engineering science at first and the integration of STEM and non-STEM elements as well (Kong H.B, 2011).

The "Big E" strategy led to CDIO (Conceive, Design, Implement, Operate) approach in the Department of Aerospace Engineering (Crawley, 2001) and had a profound influence on engineering education: Olin College of Engineering came up with "Engineering2.0" demanding new qualities for the next generation of problem solvers (Miller R.K., 2017); The STEM connector (2014) conducted a survey on the demand-side requirements and provided new capability platforms of the tomorrow's STEM 2.0 jobs; MIT, the forerunner, launched New Engineering Education

Transformation (NEET) in 2016, an interdepartmental project-centric academic program, aiming to equip students with the NEET ways of thinking, including computational thinking, and getting ready for complex, highly networked systems and higher levels of automatic machines (Crawley, 2018). Above all, “Engineering with a Big E” provides a fundamental reference for our study. Computing in engineering education is not only about studying computer science, it’s all about engineering practices with computational approaches and ways of thinking.

3. The Framework of CT for Big E (CT-ENG)

Computation thinking is a broad term that encompasses a set of concepts, techniques and skills. In this section, the study will deconstruct and define CT in the context of “engineering with Big E” (CT-ENG) as follows: We firstly draw on multiply materials including standard documents, reports, and other scholarly literature to identify the core elements of CT-ENG. We then conduct face to face semi-structured in-depth interviews with 19 professionals and 5 human resources executives from 11 enterprises and institutes. After two rounds of revisions, we formulate a framework for CT-ENG based on these four elements (See Fig.1):

- **Digital Literacy:** Understand the basic functions and terminology related to computer hardware, software, information systems, and communication device; identify, organize, analyze and visualize digital information and data; and computational culture or digital security awareness.

- **Modeling and Simulation Skills:** Activities including understanding a concept with computational models; using, assessing and testing models to find a solution; building or extending existing models or creating new models on a computational device; structured thinking, simulation thinking and etc.

- **Complex Problem Solving Skills:** Working with multiple layers of abstraction and understanding the relationships among the different layers; decomposing large complex problems into manageable modular subtasks that supports parallel execution; algorithm design and solution implementation.

- **Digital Leadership:** To initiate and guide computation-related innovation. In other words, those essential soft skills or ways of thinking for a digital age, including system thinking, interdisciplinary or integration skills, innovation and entrepreneurship, multi-cultural teamwork and collaboration skills.

Figure 1. The Framework of CT-ENG

III. Case Study: The Robotics Class of Zhejiang University

1. Research Design

(1) Research Method

This research adopts the single case study method. Case study research design is an in-depth practical investigation of a current event in the actual context (Yin, 2009). According to (Siggelkow,2007; Gaya H.J& Smith E.E.,2016), a single case study may nonetheless provide valuable insights to test theories, as long as the case study possesses the relevant attributes needed to meet the study objectives.

(2) Research Objectives and Objects

The primary objective of this single case study is to investigate how to integrate computing into engineering curriculum and cultivate computational thinking-enabled engineers. Robotics is usually seen as an interdisciplinary activity drawing mostly on mechanics, electrical, sensing, control and artificial intelligence. It thus provides opportunities to integrate programming, engineering design, mathematics, and all areas that benefit from computational thinking (Shoop R, et al., 2016). That's why the article chooses the robotics program as a case to rethink the relationship between computing and engineering education.

This paper highlights practices and experience of the Robotics Class at Chu Kochen Honors College in Zhejiang University for the following reasons:

1) Zhejiang University (ZJU) is one of China's top higher education institutions, as well as one of its oldest. The Faculty of Engineering (FE) and Faculty of Information Technology (FIT) in ZJU, comprise of fourteen colleges/schools and one department, with more than 11948 undergraduates and 8571 graduate students enrolled now. ZJU has been deeply dedicated to nurturing engineering innovators and future leaders, and thus its experience in educational reform, to some extent, possibly reveals the future trends in engineering education in the digital era.

2) Chu Kochen Honors College, named after the former president of ZJU, has been exploring innovative and prospective education models to cultivate elites (5~6% of total enrollment) since it established in 1984. The college which is neither affiliated with Faculty of Engineering, nor operated by Faculty of Information Technology, provides an integrated approach to a holistic program teaching computational thinking in the context of "Big E", overcoming the barriers to interdisciplinary education.

3) The Robotics Class established in 2016 is one of the experimental programs in Chu Kochen Honors College, and has successfully developed a flexible educational ecosystem (See Fig.2). But before that, engineering schools and departments have

their own robotics-related courses and programming courses separately. How to change discipline-oriented courses into project-based computing-aided engineering activities? The experience and lessons learned in the process of educational reform in the Robotics Class is thought to have universal significance.

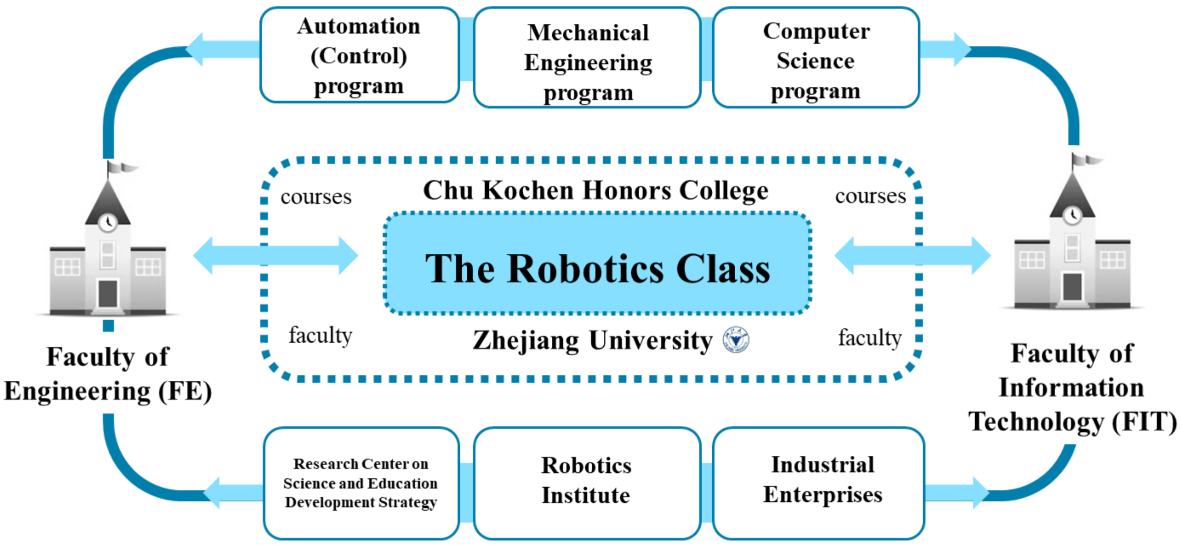

Figure 2. Educational Ecosystem of the Robotics Class

(3) Data Collection

Systematic integration of quantitative and qualitative data collection methods increase the credibility of findings when information from different data sources converges and can also deepen the understanding of the program, its effects and context (Peersman, 2014). Data for this study is collected from various sources, which include:

1) Face-to-face personal interviews with teachers from Faculty of Engineering and Faculty of Information Technology, and also administrative staff from Kochen Honors College. Those semi-structured interviews offer plenty of first-hand information on the transformations and operations of the Robotics Class;

2) Focus group discussion with students in the Robotics Class. Eight participants from different grades are encouraged to reflect openly and informally on their degree program. The discussion lasts for 60 minutes, allowing us to explore how do students learn computational thinking and what are their attitudes on the learning experiences.

3) Documental revision on programs, syllabus and other second-hand materials. The authors analyze computational components in the curriculum of the Robotics Class by examining distribution of credits scores; and conduct a comparison of the robotics programs pre-/post- transformation, figuring out the approaches to embed computing into curriculum for non-CS engineering majors.

2. Computing Integrated into Program

(1) Program Overview

The program of the Robotics Class consists of five educational modules as follows: General Education, Professional Education, Practical Education, Capstone and other elective courses. Computing-related courses are integrated into each educational module. Students are encouraged to engage in the computing-aided engineering design projects since first year, allowing them to understand what engineering is and how does computational thinking work in the problem-solving process at an early stage of learning (see Table 1).

Table 1. Program of the Robotics Class

Curriculum Module	Course Category	Credits Scores		Computing-related Courses
General Education	Political Courses	14	69.5	*Eg. Fundamentals of Programming*
	Military Courses	5.5		
	Foreign language	6		
	Computer Science	5		
	Natural Science	27		
	Innovation and Entrepreneurship	1.5		
	General Elective Courses	10.5		
Professional Education	Foundation Courses	26.5	75	*Eg. Numerical Methods in Engineering* *Modeling and Simulation on the Mechatronic Systems*
	Core Courses	20		
	Elective Courses	28.5		
Practical Education	Projects, Competitions	15	15	*Eg. International Robotics Competition* *Enhanced Training of Robotics Technology*
Capstone	Capstone	8	8	*Engineering Design Project*
Elective Course	Extracurricular Practice, International Exchange	6	6	*Eg. Fieldwork in Robotics Institute*
SUM	/	173.5	173.5	/

(2) Curriculum Transformation

The Robotics Class in ZJU originated from several robotics-related courses and some experimental programs separately in different engineering and information technology schools: Automatics Program (Robotics) in School of Control Science and Engineering has been exploring the systematic program in Robotics since 2002. Meanwhile, School of Mechanical Engineering and College of Computer Science and Technology also offer courses embedding the fundamental engineering principles and computing tools for Robotics.

The problem is that most of the courses above are discipline-oriented, so simply mixing them up can hardly be integrated into a holistic program. According to the interviews with administrative staff, teachers represented for different engineering schools and computer science and technology are fighting to in charge of the robotics program. Considering the institutional barriers and disciplinary differences, Zhu Kozhen Honor College in ZJU, which is not affiliated with Faculty of Engineering or Faculty of Information Technology, reconstructs the courses into project-based computing-aided engineering activities. We conduct pre-post comparisons and illustrate the differences in computing between Automatics Program (Robotics) in School of Control Science and Engineering and the Robotics Class in Chu Kochen College (Fig 3 &Table 2).

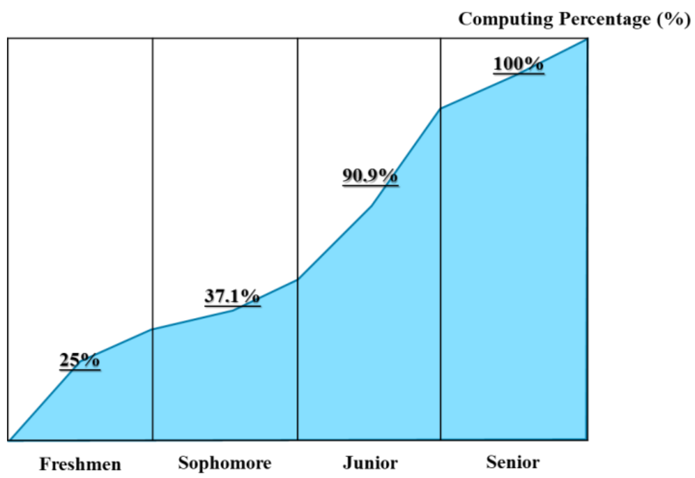

Figure 3. Computing in the Curriculum (percentage of credit scores in each year)

Table 2. The pre-post Comparisons between Two Robotics Programs

Categories	Program in 2018	Program in 2013
Computing in **General Education**	**5/53=9.4%** *Fundamentals of Programming/ Program Design*	
Computing in **Professional Education**	**21.5/53=40.6%** ↓ Eg. *Object-Oriented Programming* *Principles of Embedded System* *Numerical Methods in Engineering* *Modeling and Simulation on the Mechatronic Systems* *Principle of Automatic Control*	**18.5/35.5=52.1%** *Eg. Microcomputer Principle and Interface Technology* *Control Instruments & Computer* *Process Control Engineering*
Computing in **Practical Education&** **Capstone in CT**	**26.5/53=50%** ↑ *Eg. Introduction and Practice of Robotics and Artificial Intelligence* *Design and Practice of Computer Control System* *Innovation and Practice in Robotics*	**12/35.5=33.8%** *Capstone*
Computing in **Compulsory Curriculum**	**53/103=51.5%** ↑	**35.5/98=36.2%**

NOTE: The military, political courses and other elective courses are excluded here, and this table compares the proportion of computing in compulsory curriculum.

3. The Cultivation of Computational Thinking

The program of the Robotics Class retains some essential discipline-oriented courses, helping students to get access to basic proficiency with digital tools and computing methods, which refers to Digital Literacy and Modeling and Simulation Skills in CT-ENG: 1) There are courses mostly offered in the School of Computational Science and Engineering, such as *object-oriented Programming*, *data structure and algorithm analysis*, *Artificial Intelligence* and *Virtual Reality Technology*; 2) Computational thinking can be embedded into traditional engineering courses with the prevalent application of modeling and simulation, such as *Robotics and Programmable Controllers*, *Electromechanical Systems Design* and *Principle of Automatic Control*.

Besides that, the large majority of the courses in the Robotics Class are infused with project-based computing-aided engineering activities, creating an open and innovative learning environment in the digitally sophisticated world: 1) Complex Problem

Solving Skills and Digital Leadership in CT-ENG are set as the educational goals and related instructive activities and evaluation standards are designed into these interdisciplinary courses, such as *Introduction and Practice of Robotics*, *Innovation and Practice in Robotics*; 2) Robotics competitions (such as *RoboCup, IDC Robocon, ROBOTAC*) and internships in technology companies and laboratories can also offer chances to think computationally in engineering practices.

Table 3. CT Components in Courses

Education Model	Examples	Digital Literacy	Modeling and Simulation	Complex Problem Solving	Digital Leadership
Discipline-Oriented Courses	*Fundamentals of Programming*	√			
	Modeling and Simulation on the Mechatronic Systems		√		
Project-Based Engineering Activities	*Innovation and Practice in Robotics*			√	√
	Robotics Competitions	√	√	√	√

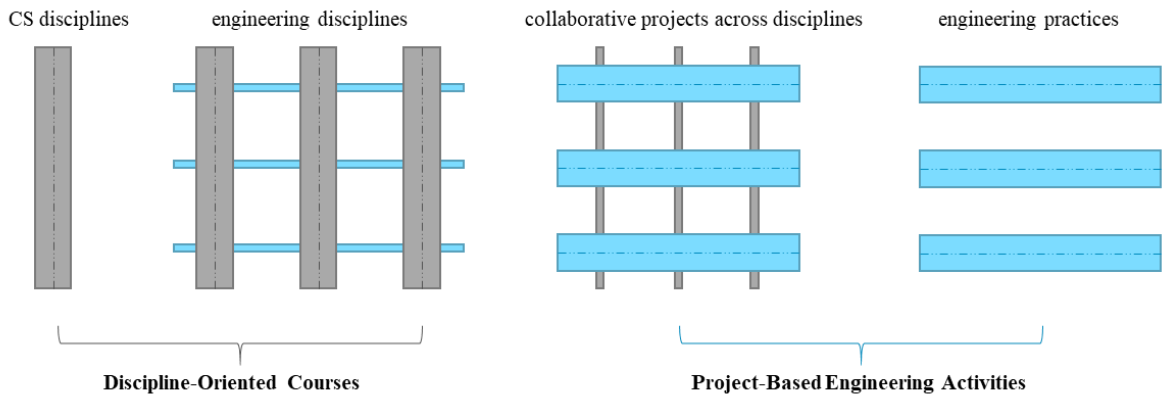

Fig 4. CT Components in Courses

To be specific, we introduce the course *Innovation and Practice in Robotics* and further explicit the integrated approach to computational thinking in the context of G"Big E": This course has only one formal lecture, most of the instructive activities are engineering practices and discussions. Students are required to choose a cutting-edge topic in robotics at the beginning of the term, and to apply digital resources creatively, using computing tools to solve problems innovatively. For example, one of the students we interviewed in the focus group discussion shares his project: *Rehabilitation Training Robot based on Human-Computer Interaction* (See Fig 5). The project integrates multidisciplinary knowledge in mechanical engineering, sensing and automation, information technology, and life sciences, as well as human emotions and cognition. Students experience a real and complete engineering lifecycle through solving complex engineering problems in a human-computer interaction context, so they can better be prepared for the real-world challenges of an increasingly digital workplace.

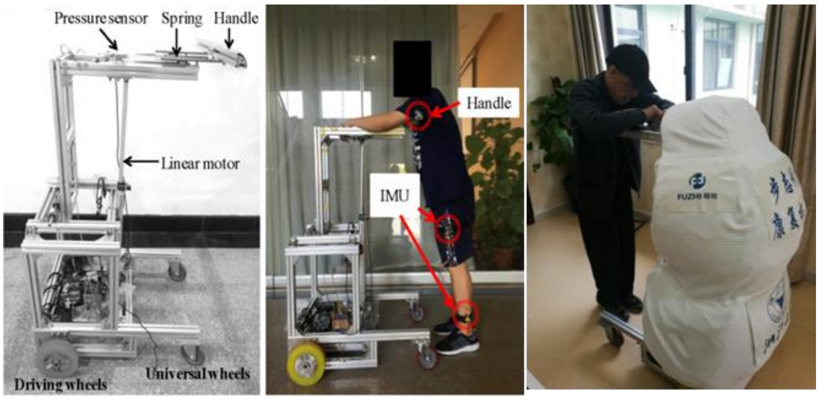

Fig 5. Rehabilitation Training Robot Designed by Students in the Robotics Class

IV. Conclusion, Limitation and Future Work

Computational Thinking (CT) is essential to survival in the digital world for engineers. Robotics programs provide learning environments for acquiring core computational thinking skills. This study proposes a framework of computational thinking in the context of complex engineering systems (CT-ENG), based on interviews with engineering specialists. Considering how to teach computational thinking in the context of engineering is still challenging, this paper introduces the Robotics Class of Zhejiang University in China and concludes with an integrative approach to the education reformation. In this research, we emphasize that computational thinking is not about the depth knowledge of computer science, it is about “think like a computer scientist”, including thinking logically, systematically and innovatively.

This paper is only a first attempt to summarize the practices and experience for equipping engineering students with CT skills. More studies are needed in order to validate the framework of CT-ENG and investigate the formation mechanism of computational thinking.

Acknowledgments

Research reported in this paper is supported by Chinese Academy of Engineering under Program No. 2018-XY-051, Chinese Ministry of Education under Humanities and Social Sciences Research Foundation No.18YJC630197, and National Science Foundation in Zhejiang Province under Program No.LQ19G030009. The content is solely the responsibility of the authors and does not necessarily represent the official views of the academies or the Foundation.

References

[1] Beheshti E, et al. Computational Thinking in Practice: How STEM Professionals Use CT in Their Work[C].American Education Research Association Annual Meeting[A],2017

[2] Chen G.L. Computational Thinking[J]. Communications of the CCF, 2012(1): 31-34 (in Chinese)

[3] CSTA& ISTE. Computational Thinking in K–12 Education leadership toolkit[Z],2011

[4] Council on Competitiveness. U.S. Manufacturing – Global Leadership Through Modeling and Simulation White paper[R],2009

[5] Crawley E F, et al. Rethinking Engineering Education: The CDIO Approach [M]. New York, Springer, 2007: 1-5

[6] Crawley E F, et al. Redesigning Undergraduate Engineering Education at MIT – the New Engineering Education Transformation (NEET) initiative[C]. 2018 ASEE Annual Conference & Exposition[A], 2018

[7] Dijkstra E W. On the Teaching of Programming, ie. on the Teaching of Thinking [A]. Language hierarchies and interfaces[C].Springer, Berlin, 1976: 1-10

[8] Gaya H.J& Smith E.E. International Journal of Business Management and Economic Research (IJBMER) [J]. 2016,7(2): 529-538

[9] Gross S, et al. Fostering computational thinking in engineering education: Challenges, examples, and best practices[C]. Global Engineering Education Conference (EDUCON) [A]. IEEE, 2014: 450-459

[10] Hacker M. Integrating computational thinking into technology and engineering education [J]. Technology and Engineering Teacher, 2018, 77(4): 8-14.

[11] Kalelioglu F, et al. A framework for computational thinking based on a systematic research

review [J]. Baltic Journal of Modern Computing,2016,4(3): 583
[12] Kong H.B. The Evolution and Transformation of the Knowledge Body of Engineering Disciplines [J]. Research in Higher Education of Engineering, 2011(04): 27-34 (in Chinese)
[13] Korkmaz Ö, et al. A validity and reliability study of the Computational Thinking Scales (CTS)[J]. Computers in Human Behavior, 2017, 72: 558-569
[14] Magana A J, et al. Introducing discipline-based computing in undergraduate engineering education [J]. ACM Transactions on Computing Education (TOCE),2013,13(4): 16
[15] Malyn-Smith J& Lee I. Application of the occupational analysis of computational thinking-enabled STEM professionals as a program assessment tool [J]. Journal of Computational Science Education, 2012, 3: 2-10
[16] Miller R K . Building on Math and Science: The New Essential Skills for the 21st-Century Engineer: Solving the Problems of the 21st Century Will Require That Engineers Have a New Set of Skills and Mindsets [J]. Research-Technology Management, 2017
[17] Mohtadi C, et al. Why integrate computational thinking into a 21st century engineering curriculum?[C]. Proceedings of SEFI Annual Conference 2013[A],2013
[18] Moses J. Engineering with a Big E: integrative education in engineering: long range plan for the School of Engineering, 1994-1998[M]. School of Engineering, Massachusetts Institute of Technology, 1994
[19] Peersman. Overview: Data collection and analysis methods in impact evaluation [EB/OL], http://www.unicef-irc.org/KM/IE/,2014
[20] Repenning A, et al. Scalable game design and the development of a checklist for getting computational thinking into public schools[C].Proceedings of the 41st ACM technical symposium on Computer science education [A]. ACM, 2010: 265-269
[21] Shoop R, et al. Can computational thinking practices be taught in robotics classrooms[C]//International Technology and Engineering Education Conference[A].2016
[22] Siggelkow N. Persuasion with case studies. Academy of Management Journal [J].2007, 50(1): 20-24.
[23] STEM connector's Innovation Task Force. STEM 2.0-An Imperative For Our Future Workforce[EB/OL], https://www.stemconnector.com/publication/stem-2-0,2014-10
[24] Swaid S I. Bringing computational thinking to STEM education [J]. Procedia Manufacturing, 2015, 3: 3657-3662
[25] Vest, C. Our Revolution [J]. ASEE Prism, 1994, 3(9): 40
[26] Weintrop D, et al. Defining computational thinking for science, technology, engineering and math[C]. American Educational Research Association Annual Meeting[A], Pennsylvania, 2014
[27] Wenchong, et al. Computational Features of the Thinking and the Thinking Attributes of Computing: On Computational Thinking [J]. JSW, 2014, 9(10): 2507-2513
[28] Wing J M. Computational Thinking [J]. Communications of the ACM,2006,49(3): 33-35
[29] Wu S.M, Wang P.M. The reform direction of engineering graduate education in the 21st century [J]. Academic Degrees & Graduate Education, 1995(06): 10-13 (in Chinese)
[30] Yin R K. Case study research: Design and methods [J]. Journal of Advanced Nursing, 2010, 44(1):108-108.
[31] Zhan D.C., Nie L.S. The path to computational thinking and university computing curriculum reform [J]. China University Teaching, 2013(2):56-60 (in Chinese)